



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/775,763	02/10/2004	Mycong-Cheol Shin	8836-240 (IB14008-US)	8723
22150 7590 01/09/2007 F. CHAU & ASSOCIATES, LLC 130 WOODBURY ROAD WOODBURY, NY 11797			EXAMINER BAKER, STEPHEN M	
			ART UNIT 2133	PAPER NUMBER
SHORTENED STATUTORY PERIOD OF RESPONSE 3 MONTHS			MAIL DATE 01/09/2007	DELIVERY MODE PAPER

**Please find below and/or attached an Office communication concerning this application or proceeding.**

If NO period for reply is specified above, the maximum statutory period will apply and will expire 6 MONTHS from the mailing date of this communication.

**Office Action Summary**

Application No.

10/775,763

Applicant(s)

SHIN, MYEONG-CHEOL

Examiner

Stephen M. Baker

Art Unit

2133

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

- 1) ☒ Responsive to communication(s) filed on 21 November 2005.
- 2a) ☐ This action is FINAL. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

- 4) ☒ Claim(s) 1-20 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-20 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

**Application Papers**

- 9) ☒ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 10 February 2004 is/are: a) ☐ accepted or b) ☒ objected to by the Examiner.
- Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
- Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some \* c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

\* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☒ Information Disclosure Statement(s) (PTO/SB/08)  
Paper No(s)/Mail Date 112105.
- 4) ☐ Interview Summary (PTO-413)  
Paper No(s)/Mail Date. \_\_\_\_\_.
- 5) ☐ Notice of Informal Patent Application
- 6) ☐ Other: \_\_\_\_\_.

## DETAILED ACTION

### *Drawings*

1. Figures 1-4 should be designated by a legend such as --Prior Art-- because only that which is old is illustrated. See MPEP § 608.02(g).
2. Corrected drawings in compliance with 37 CFR 1.121(d) are required in reply to the Office action to avoid abandonment of the application. The replacement sheet(s) should be labeled "Replacement Sheet" in the page header (as per 37 CFR 1.84(c)) so as not to obstruct any portion of the drawing figures. If the changes are not accepted by the examiner, the applicant will be notified and informed of any required corrective action in the next Office action. The objection to the drawings will not be held in abeyance.

### *Specification*

3. The disclosure is objected to because of the following informalities:  
The specification is not written entirely in idiomatic English and contains numerous minor errors. Suggested corrections for paragraphs 000-0074 are provided below, correction of the remaining informalities is left to applicant:

#### ABSTRACT:

A processor on which a software-based interleaver is run is provided. The interleaver generation is split into two parts to reduce the overhead time of interleaver changing. First, preprocessing prepares seed variables, requiring a small memory. Second, on-the-fly address generation generates interleaved address addresses through simple adding and subtracting operations using the seed variables.

**BRIEF SUMMARY:****TECHNICAL FIELD**

[0001] The present invention relates to a data processing system, and more particularly to a turbo decoder and a turbo interleaver.

**BACKGROUND OF THE INVENTION**

[0002] Data signals, in particular those transmitted over a typically hostile RF interface (communication channel), are susceptible to error (channel noise) caused by the interface. Various methods of error correction coding have been developed in order to minimize the adverse effects that a hostile interface has on the integrity of communicated data. This is also referred to as lowering the Bit Error Rate (BER), which is generally defined as the ratio of incorrectly received information bits to the total number of received information bits. Error correction coding generally involves representing digital data in ways designed to be robust with respect to bit errors. Error correction coding enables a communication system to recover original data from a signal that has been corrupted.

[0003] ~~Error~~ Two types of error correction code are a convolutional convolutional code, and a parallel concatenated convolutional convolutional code (so called turbo code). A ~~convolutional~~ convolutional code transforms an input sequence of bits into an output sequence of bits through the use of a finite-state-machine, where additional bits are added to the data stream to allow for provide error-correction capability. In order to increase error-correction capability, the amount of additional bits added and the amount of memory ~~preset~~ present in the finite-state-machine must be increased, which increases decoding complexity.

[0004] In the turbo coding system, a block of data may be encoded with a particular coding method resulting in systematic bits and two sets of parity bits. Additionally For generating the second set of parity bits, the original block of input data ~~may be~~ is rearranged with an interleaver and then encoded with the same method as that applied to the original input data in generating the first set of parity bits. Encoded data (systematic bits and parity bits) are combined in some manner to form a serial bit stream and transmitted through the communication channel to a turbo decoding system. Turbo decoding systems ~~operates~~ operate on noisy versions of the systematic bits and the two sets of parity bits in two decoding stages to produce an estimate of the original message bits. The turbo decoding system uses an iterative decoding ~~algorithms~~ algorithm and ~~consist~~ consists of interleaver and deinterleaver stages, individually matched to constituent decoding stages. The decoding stages of the turbo decoding system

Art Unit: 2133

uses may use the BCJR algorithm, which was originally invented by Bahl, Cocke, Jelinek, and Raviv (~~hence the name~~) to solve a maximum a posteriori probability (MAP) detection problem. The BCJR algorithm is a MAP decoder decoding algorithm in that it minimizes the bit errors by estimating the a posteriori probabilities of the individual bits in a code word; ~~to~~. To reconstruct the original data sequence, the soft outputs of the BCJR algorithm are hard-limited. The decoding stages exchange with each other the obtained soft output information and iteration of decoding is ceased when a satisfactory estimate of the transmitted information sequence has been achieved.

[0005] As the turbo code has extremely impressive performance which is very close to Shannon capacity limits, the 3G mobile radio systems such as W-CDMA and cdma2000 have adopted them turbo codes for channel coding.

[0006] 3G wireless systems support the a variable bit rate, which may result in full reconstruction of the turbo interleaver at every 10 ms or 20 ms frame. Accordingly, generating the whole interleaved address pattern at once consumes much time and requires a large-sized RAM to store the pattern.

[0007] Accordingly, a high speed turbo interleaver which can support a variable bit rate and which does not affect the performance of the turbo coder is required.

[0008] As is well-known, W-CDMA and cdma2000 transmission schemes are different in coding rate and interleaver interleaving. For example, the coding rate of W-CDMA is can be 1/2, 1/3, 1/4 or 1/5 but the coding rate of cdma2000 is 1/3, and the frame size of the W-CDMA is one of twelve numbers 378, 370, 762, . . . , and 20730, but that of the cdma2000 is an arbitrary integer between 40 and 5114, and the row size of the block interleaver in W-CDMA is 32 (8-14 of them are unused) but that of the cdma2000 is can be 5, 10, or 20.

[0009] Accordingly, flexible and programmable decoders are required for 3G communication because global roaming is recommended between different 3G standards and the frame size may change on a frame base.

## SUMMARY OF THE INVENTION

[0010] Embodiments of the present invention provide an interleaver. The interleaver comprises a preprocessing means for preparing seed variables and an address generation means for generating interleaved address addresses, using the seed variables, on the fly. The seed variables are in forms the form of column vectors whose number of elements is equal to that the number of the rows of the two dimensional block interleaver. Consequently the number of seed variables is less than the size of the data block data. If the a generated

interleaved address is larger than the size of the data block data, the generated interleaved address is discarded.

[0011] In some embodiments, the seed variables include a base column vector, an increment column vector and a cumulative column vector. The number of elements of all three column vectors is equal to the number of ~~row~~ rows of the interleaver block. The cumulative column vector is updated by adding the increment vector to the old cumulative column vector, ~~after then~~ interleaved addresses for one column is are generated by adding the base column vector and the cumulative column vector. When updating the cumulative column vector, if elements of the updated cumulative column vector are larger than the number of ~~column of~~ columns in the data block, ~~the elements of the updated cumulative column vector is subtracted~~ are reduced by the number of ~~the column of~~ columns in the data block.

[0012] Elements of the base column vector and the increment column vector are arranged by-inter-row permuted permutation in advance.

[0013] Embodiments of the present invention provide a turbo decoding system. The turbo decoding system comprises an interleaver comprising a preprocessing means for preparing seed variables and an address generation means for generating interleaved address using the seed variables, an address queue for storing ~~the a~~ generated interleaved address equal to or smaller than the interleaver size, an SISO decoder performing recursive decoding and calculating a log likelihood ratio, and an LLR memory connected to the SISO decoder and storing the log likelihood ratio, wherein the SISO decoder accesses the input data and the log likelihood ratio alternately in a sequential order and in an interleaved order using the generated interleaved address.

[0014] In some embodiments, the generated interleaved address is reused as a write address for writing the log likelihood ratio outputted from the SISO decoder into the LLR memory.

[0015] Embodiments of the present invention provide a turbo decoding system comprising a processor for generating interleaved addresses and controlling hardware blocks, an address queue for storing the generated interleaved addresses, a buffer memory block including an LLR memory for storing a log likelihood ratio and a plurality of memory blocks for storing soft inputs, an SISO decoder connected to the buffer memory block, the SISO decoder including an ACSA network for calculating the log likelihood ratio recursively from soft inputs and the log likelihood provided by the LLR memory and a plurality of memory blocks for storing intermediate results of the ACSA network.

[0016] In some embodiments, the processor ~~preparing~~ prepares seed variables when the interleaver structure changes due to the a change of the coding standard or bit rate, and generates the interleaved addresses column by column using the seed variables by simple add and subtract operations when the interleaved addresses are required.

[0017] In some embodiments, the SISO decoder supports a Viterbi decoding mode. In the Viterbi decoding mode, the ACSA network performs a Viterbi recursion, the LLR memory stores traceback information outputted by the ACSA network, the processor ~~processes~~ performs traceback from the traceback information read from the LLR memory, and one of the ~~memory~~ memories of the SISO decoder stores a path metric outputted by the ACSA network.

[0018] In some embodiments, the processor is a single-instruction and multiple-data (SIMD) processor. Preferably, the SIMD processor includes five processing elements, and wherein one of five processing elements controls the other four processing elements, processes scalar operation, and fetches, decodes, and executes instructions including control and multi-cycle scalar instructions, and wherein the other four processing elements only execute SIMD ~~instruction~~ instructions.

[0019] Embodiments of the present invention provide an interleaving method for rearranging a data block in a data communication system. The interleaving method comprises preparing seed variables and generating interleaved addresses on the fly using the seed variables.

#### DRAWING DESCRIPTION:

#### BRIEF DESCRIPTION OF THE DRAWINGS

[0020] Other features of the present invention will be more readily understood from the following detailed description of the invention when read in conjunction with the accompanying drawings, in which:

[0021] FIG. 1 illustrates a basic turbo-encoding system.

[0022] FIG. 2 illustrates a typical eight-state RSC encoder of FIG. 1.

[0023] FIG. 3 shows a block diagram of the turbo decoding system.

[0024] FIG. 4 shows the extrinsic form of the turbo decoding system of FIG. 3.

[0025] FIG. 5 schematically shows a block diagram of a time-multiplex turbo decoding system according to an embodiment of the present invention.

[0026] FIGS. 6A to 6C illustrate a simple example of a prunable block interleaver with interleaver size  $N = 18$  according to a conventional interleaving technique.

[0027] FIGS. 7A and 7B illustrate a simple example of a prunable block interleaver with interleaver size  $N = 18$  according to the present invention.

[0028] FIG. 8 ~~illustrate~~ illustrates a more specified block diagram of turbo decoding system of FIG. 5 in turbo decoding mode.

[0029] FIG. 9 ~~illustrate~~ illustrates more specified detailed block diagram of the turbo decoding system of FIG. 5 in Viterbi decoding mode.

[0030] FIG. 10 schematically shows detailed in further detail the ACSA network and related memory blocks of FIG. 8.

[0031] FIG. 11 shows an ACSA unit contained in the ACSA A section 1022 of FIG. 10 for calculating the forward metric  $A_k(s)$  ~~of FIG. 10~~.

[0032] FIG. 12 illustrates detailed SIMD processor of FIG. 8.

#### DETAILED DESCRIPTION:

#### DETAILED DESCRIPTION OF THE INVENTION

[0033] The present invention now will be described more fully hereinafter with reference to the accompanying drawings, in which typical embodiments of the invention are shown. This invention may, however, be embodied in many different forms and should not be construed as limited to the embodiments set forth herein. Rather, these embodiments are provided so that this disclosure will be thorough and complete, and will fully convey the scope of the invention to those skilled in the art.

[0034] Before proceeding to describe the embodiments of the present invention, a typical turbo coding system will be described with reference to FIGS. 1 to 4 for better understanding of the present invention.

[0035] FIG. 1 illustrates a basic turbo-encoding system 100 and FIG. 2 illustrates a typical eight-state RSC encoder 102, 106 of FIG. 1.



Art Unit: 2133

[0036] The encoder of the turbo coding system consists of two constituent systematic encoders 102, 106 joined together by means of an interleaver 104. Input data stream  $u$  is applied directly to first encoder 102, and the interleaved version of the input data stream  $u$  is applied to second encoder 106. The systematic bits (i.e., the original message bits)  $x^s$  and the two sets of parity bits  $x^{p1}$  and  $x^{p2}$  generated by the two encoders 102, 106 constitute the output of the turbo encoding system 100 and are combined by a multiplexing means 108 to form a serial bit stream and that is transmitted over the communication channel. Before transmission, puncturing may be performed if necessary.

[0037] The constituent encoder 102, 106 of the turbo encoding system is a recursive systematic convolution (RSC) code encoder, where one or more of the tap outputs in the ~~sift-resister~~ shift-register D1 ~ D3 back to the input for obtaining better performance of the overall turbo coding strategy.

[0038] FIG. 3 shows a block diagram of the turbo decoding system 300. The turbo decoder 300 operates on a noisy version of the systematic bits  $y^s$  and noisy ~~version of the two set of~~ parity bits  $y^{p1}$  and  $y^{p2}$ . The turbo decoding system 300 uses an iterative decoding ~~algorithms~~ algorithm and consist of interleaver 304 and deinterleaver 308, 310 stages, individually matched to constituent decoding stages 302, 306. The systematic bits  $y^s$  and first set of parity bits  $y^{p1}$  of turbo encoded data are applied to the first SISO (Soft-Input-Soft-Output) decoder 302. Additionally, the deinterleaved version of the metrics output ~~of from~~ the second SISO decoder 306 is are fed back to the first SISO decoder 302. The metrics output ~~of from~~ the first SISO decoder 302 is are applied to the second SISO decoder 306 via interleaver 304. The second set of parity bits  $y^{p2}$  is applied to the second SISO decoder 306. The output of the deinterleaver 310 is applied to hard limiter 312 which outputs a bit stream of decoded data  $u$  corresponding to the original raw data  $u$ .

[0039] As stated earlier, the metrics output ~~of from~~ f the deinterleaver 308 is are fed back to the ~~metrics~~ input of the first SISO decoder 302. Thus the turbo decoding system 300 performs the  $n$ th decoding iteration with ~~an~~-input metrics resulting from  $(n-1)$ th decoding iteration. The total number of ~~iteration~~ iterations is either predetermined, or the ~~iteration-steps~~ iterations stop if a certain stopping criterion ~~meets the qualification~~ is met.

[0040] FIG. 4 shows the extrinsic form of the turbo decoding system of FIG. 3, where I stands for interleaver, D for deinterleaver, and SISO for soft-input soft-output decoder, which may use a Log-MAP decoding algorithm, or Max-Log-MAP, etc.

Art Unit: 2133

[0041] The first decoding stage produces a soft estimate  $\Lambda(u_k)$  of a systematic bit  $u_k$  expressed as a log-likelihood ratio

$$\Lambda_1(u_k) = \log \{P(u_k=1|y^s, y^{p1}, \Lambda_{2e}(u)) / P(u_k=0|y^s, y^{p1}, \Lambda_{2e}(u))\}, k = 1, 2, \dots, N \quad (1)$$

[0042] where  $y^s$  is the set of noisy systematic bits,  $y^{p1}$  is the set of noisy parity bits generated by the first encoder 302, and  $\Lambda_{2e}(u)$  is the extrinsic information about the set of message bits  $u$  derived from the second decoding stage and fed back to the first stage.

[0043] Hence, the extrinsic information about the message bits derived from the first decoding stage is

$$\Lambda_{1e}(u) = \Lambda_1(u) - \Lambda_{2e}(u) \quad (2)$$

[0044] where  $\Lambda_{2e}(u)$  is to be defined.

[0045] Before application to the second decoding stage, the extrinsic information  $\Lambda_{1e}(u)$  is reordered to compensate for the interleaving introduced in the turbo encoding system 100. In addition, the noisy parity bits  $y^{p2}$  generated by the second encoder 106 are used as another input. Thus Then, by using BCJR algorithm, the second decoding produces a more refined soft estimate of the message bits  $u$ .

[0046] This estimate is de-interleaved to produce the total log-likelihood ratio  $\Lambda_2(u)$ . The extrinsic information  $\Lambda_{2e}(u)$  fed back to the first decoding stage is therefore

$$\Lambda_{2e}(u) = \Lambda_2(u) - \Lambda_{1e}(u) \quad (3)$$

[0047] where  $\Lambda_{1e}(u)$  is itself ~~defined~~ defined by equation (2), and  $\Lambda_2(u)$  is the log-likelihood ratio computed by the second decoding stage. Specifically, for the  $k$ th element of the vector  $u$ , where we have

$$\Lambda_2(u_k) = \log \{P(u_k=1|y^s, y^{p2}, \Lambda_{1e}(u)) / P(u_k=0|y^s, y^{p2}, \Lambda_{1e}(u))\}, k = 1, 2, \dots, N \quad (4)$$

[0048] Through the application of  $\Lambda_{2e}(u)$  to the first decoding stage, the feedback loop around the pair of decoding stages is thereby closed. Note that although in actual fact the set of noisy systematic bits  $y^s$  is only applied to the first decoder 302 in FIG. 3, by formulating the information flow in the symmetric extrinsic manner depicted in FIG. 4 we find that  $y^s$  is, in fact, also applied to the second decoding stage.

[0049] An estimate of message bits  $\mathbf{u}$  is computed by hard-limiting the log-likelihood ratio  $\Lambda_2(\mathbf{u})$  at the output of the second decoding stage, as shown by

[0050]  $\mathbf{u} = \text{sgn}(\Lambda_2(\mathbf{u}))$ , where the signum function operates on each element of  $\Lambda_2(\mathbf{u})$  individually. To initiate the turbo decoding algorithm, we simply set  $\Lambda_{2e}(\mathbf{u}) = 0$  on the first iteration of the algorithm.

[0051] Now a turbo decoding system of the present invention will be described. Embodiments of the present invention provide a multi-standard turbo decoding system with a processor on which a software interleaver is run. Particularly, the present invention provides a turbo decoding system with a configurable hardware SISO decoder and a programmable single-instruction and multiple-data (SIMD) processor performing flexible tasks such as interleaving. ~~Software~~ The software turbo interleaver is run on the SIMD processor. The decoding system of the present invention can also support the Viterbi decoding algorithm as well as a turbo decoding algorithm such as the Log-MAP algorithm and the Max-Log-MAP algorithm.

[0052] The processor generates interleaved addresses for turbo decoder, supporting multiple 3G wireless ~~standard~~ standards at the speed of the hardware SISO decoder and changing the interleaver structure (i.e., frame size and bit rate) at in a very short time with a small memory. To hide the timing overhead of interleaving changing, the interleaved addresses generation is split into ~~two~~ two parts, pre-processing and incremental on-the-fly generation. ~~Pre-processing~~ The pre-processing part prepares a small number of seed variables and the incremental on-the-fly generation part generates interleaved ~~address~~ addresses based on the seed variables on the fly. When bit rate changes, the processor carries out ~~only~~ the pre-processing part to prepare only a small number of seed variables hence requiring a short time and a small memory. Whenever the interleaved address sequence is required, the processor generates the interleaved ~~address~~ addresses using the seed variables. This splitting method reduces the timing overhead of the interleaver and requires only a small memory to save the ~~seeding~~ seed variables.

[0053] FIG. 5 schematically shows a block diagram of a time-multiplex turbo decoding system according to an embodiment of the present invention.

[0054] Turbo decoding system 500 of the present invention comprises an SISO decoder 502, processor 504 on which software interleaver is run and a  $\Lambda_e$  memory 506 for storing an extrinsic log-likelihood ratio (LLR) as illustrated in FIG. 5. Data are sequentially stored in  $\Lambda_e$  memory 506 as they are always read and written in-place. For each iteration, data are accessed in a sequential order for the oddth SISO decoding and in an interleaved order for the eventh SISO decoding. Namely, in the oddth (first) SISO decoding, SISO decoder 502

receives data in a sequential order from the  $\Lambda_e$  memory 506 and ~~calculate~~ calculates a log-likelihood ratio, and the log-likelihood ratio is written into the  $\Lambda_e$  memory 506 in the sequential order. In the eventh (second) SISO decoding, SISO decoder 502 receives data in an interleaved order from the  $\Lambda_e$  memory 506 and ~~calculate~~ calculates a new log-likelihood ratio, and the new log-likelihood ratio is deinterleaved with the help of the address queue 508 and written into the  $\Lambda_e$  memory 506 in the sequential order. As shown with the dotted lines, the processor 504 provides interleaved address addresses to read data in an interleaved order. The address queue 508 saves the addresses in an interleaved order so that the addresses can be used as the write addresses for  $\Lambda_e$  memory 506 when the SISO decoder 502 produces results after its latency. Accordingly, data in an interleaved order can be deinterleaved into the sequential order, and saved in the  $\Lambda_e$  memory 506 in a the sequential order.

[0055] In addition to interleaving, the processor 504 can control the hardware blocks or interface with an external host, and processes the trellis termination and a stopping criterion during the first SISO decoding that does not need interleaved addresses. In a Viterbi decoding mode, SISO decoder 502 is repeatedly used for the Viterbi recursion. The  $\Lambda_e$  memory 506 plays the roles of the traceback memory. The processor 504 ~~processes~~ performs the traceback from the traceback information read from the  $\Lambda_e$  memory 506.

[0056] Flexible software turbo interleaver run on the processor for turbo decoder will be described. To hide the timing overhead of interleaver change, the interleaved addresses address generation is split into ~~two~~ two parts, pre-processing and incremental on-the-fly generation. ~~Pre-processing~~ The pre-processing part prepares a small number of seed variables and the incremental on-the-fly generation part generates interleaved address addresses based on the seed variables on the fly. When the interleaver size changes due to the a change of bit rate or the communication standard itself, ~~only~~ the pre-processing part prepares only a small number of seed variables, not all the interleaved address sequence. Through parallel processing using the seed variables, the processor generates interleaved addresses as fast as the hardware SISO decoding rate whenever the interleaved address sequence is required. The unit of on-the-fly address generation is a column of a block interleaver. Interleaved addresses are generated column by column, used for read addresses and stored in address queue, and then reused for write addresses for deinterleaving.

[0057] Before proceeding to describe interleaving ~~technique~~ techniques according to the embodiment of the present invention, conventional interleaving techniques will be described for better understanding of the present invention.

[0058] The turbo decoders for wireless system are based on block turbo interleavers. Although the operations and parameters vary depending on the

standards, W-CDMA and cdma2000 ~~shares~~ share the prunable block interleaver structure where the interleaver interleaver is implemented by building a mother interleaver of a predefined size and then pruning unnecessary addresses. The mother interleavers can be viewed as two-dimensional matrices, where the entries are ~~read~~ written in the matrix row by row and read out column by column. Before reading out the entries, intra- and inter-row permutations are performed.

[0059] FIGS. 6A to 6C illustrate a simple example of a typical prunable block interleaver with interleaver size  $N=18$ , where the data ~~indexes~~ are written in a matrix form. Incoming data are written into memory <sub>1</sub> in two-dimensional matrix row by row <sub>1</sub> as shown in FIG. 6A. FIG. 6B shows intra-row permutation data indexes from FIG. 6A. The intra-row permutation rule applied to this example is

$$y_{ij} = b_i + [(j + 1) \cdot q_i] \bmod 5 \text{ ---- (5),}$$

[0060] where  $y_{i,j}$  is the permuted index of the  $i$ th row and  $j$ th column,  $i$  and  $j$  are row and column indexes, respectively,  $\mathbf{b} = (b_0, b_1, b_2, b_3) = (0, 5, 10, 15)$ , and  $\mathbf{q} = (q_0, q_1, q_2, q_3) = (1, 2, 3, 7)$ . FIG. 6C shows inter-row permutation data result from FIG. 6B, which will be read out from the memory in column by column as a sequence of 17, 1, 13, 7, 2, 11, 9, . . . , 0, 10, 5. The indexes 19, 18 exceeding the range of interest are pruned.

[0061] Now an interleaving technique according to an embodiment of the present invention will be described with the example of FIGS. 6A to 6C and with reference to FIGS. 7A and 7B.

[0062] The present embodiment uses an increment vector  $\mathbf{w}$  of  $w_i = q_i \bmod 5$  instead of  $\mathbf{q}$ , and a cumulative vector  $\mathbf{x}_j$  of

$$x_{ij} = [(j + 1) \cdot q_i] \bmod 5 \text{ ---- (6).}$$

[0063] Equation (6) can be rewritten as

$$y_{ij} = b_i + x_{ij} \text{ ---- (7)}$$

[0064] and  $\mathbf{x}_j$  can be obtained recursively as

$$\begin{aligned} x_{i,j} &= [(j + 1) \cdot (q_i \bmod 5)] \bmod 5 \\ &= [(j + 1) \cdot w_i] \bmod 5 \\ &= [(jw_i \bmod 5) + w_i] \bmod 5 \\ &= (x_{i,j-1} + w_i) \bmod 5, \text{ ---- (8)} \end{aligned}$$

[0065] where  $j = 1, 2, 3, 4, 5$  and  $\mathbf{x}_0 = \mathbf{w}$ .

As  $0 \leq x_{i,j-1} < 5$  and  $0 \leq w_i < 5$ ,  $0 \leq x_{i,j-1} + w_i < 10$  and thus

$$\begin{aligned}
 x_{i,j-1} &= x_{i,j-1} + w_i - 5 && \text{if } x_{i,j-1} + w_i \leq 5, \\
 &= x_{i,j-1} + w_i && \text{otherwise} \text{ ---- (9)}
 \end{aligned}$$

[0066] According to the embodiments of the present invention, the multiplication and modulo operations of equation (6) are replaced by cheaper operation operations, multiplication by an addition and the modulo by a comparison and subtract operation.

[0067] As shown in FIG. 7A, **b**, **w**, and **x<sub>0</sub>** for the first column of the block interleaver are calculated and stored in vector resister register of the processor in the preprocessing stage. The number of elements of each vector **b**, **w**, and **x<sub>0</sub>** ~~correspond~~ corresponds to the number of elements of a column. Right side of the FIG. 7A shows that **b**, **w**, and **x<sub>0</sub>** are stored in the order of inter-row permutation such as **b**<sub>3</sub>, **b**<sub>0</sub>, **b**<sub>2</sub>, **b**<sub>1</sub> = (15, 0, 10, 5), and **w** = (**w**<sub>3</sub>, **w**<sub>0</sub>, **w**<sub>2</sub>, **w**<sub>1</sub>) = (2, 1, 3, 2) = **x<sub>0</sub>** in advance so as to free the on-the-fly generation from the inter-row permutation.

[0068] In the column by column on-the-fly address generation stage shown in FIG. 7B, the processor updates **x<sub>i</sub>** according to equation (9) and calculates the addresses based on equation (7). Calculated addresses are sent to the address queue, if they are smaller than interleaver size *N*.

[0069] Referring to FIG. 7B, interleaved addresses for first column (**y<sub>i,1</sub>**) is calculated by adding **b<sub>i</sub>** + **x<sub>0</sub>**. Since **x<sub>0</sub>** is **w**, **y<sub>i,1</sub>** is calculated by adding **b**<sub>0</sub> = (15, 0, 10, 5) + (2, 1, 3, 2) = (17, 1, 13, 7). After interleaved addresses for first column is are calculated, **x<sub>i</sub>** is updated by adding **x<sub>0</sub>** = (2, 1, 3, 2) and **w** = (2, 1, 3, 2). Thereby **x<sub>1</sub>** is set to (4, 2, 1, 4), where the third element of **x<sub>1</sub>** is 1 because (3 + 3) is larger than 5 and thus 5 is subtracted by 5. Accordingly, interleaved addresses for the second column (**y<sub>i,2</sub>**) is are calculated by adding **b<sub>i</sub>** + **x<sub>1</sub>** = (15, 0, 10, 5) + (4, 2, 1, 4) = (19, 2, 13, 7), where first element 19 is discarded since it is larger than or equal to the size *N* (=18).

[0070] As described above, the present invention requires only a small memory for storing the seed variables and performs add and subtract operations instead of modulo operation and multiplication operation operations.

[0071] The above example is a very simple one and the turbo interleavers in the real world such as W-CDMA and cdma2000 standard turbo interleavers are much more complex. However, the fundamental structure and the basic operations used in permutation rules are the same. The pseudocodes of the on-the-fly address generation for W-CDMA and cdma2000 are shown in Table 1 and Table 2, respectively.

[0072] In Table 1,  $C$  is the number of columns,  $R$  is the number of rows,  $p$  is a prime number, and  $s(x)$  is a permutation sequence, which are determined from the interleaver size  $N$  according to the specification of the W-CDMA standard. Likewise  $C$ ,  $R$ , and the binary power  $2^n$  in Table 2 are determined from  $N$  according to the cdma2000 specification. The present invention handles those values also as seed variables and calculates them in advance at the preprocessing stage.

[0073] The on-the-fly generation flows of these real-world turbo interleavers are similar to the example. They also have base column vector  $\mathbf{b}$ , increment column vector  $\mathbf{w}$ , cumulative column vector  $\mathbf{x}$ , and a certain modulo base.  $\mathbf{x}$  is updated by adding  $\mathbf{w}$  to the old  $\mathbf{x}$  value. If elements of the updated  $\mathbf{x}$  are larger than the modulo base, the those elements of the updated cumulative column vector is subtracted are reduced by the modulo base. This operation substitutes a computationally computationally expensive modulo operation. Then the interleaved addresses for one column are generated by adding  $\mathbf{b}$  and a vector that is calculated from  $\mathbf{x}$ .

TABLE 1

```

-----
0   column_counter = C-1
loop:
1    $\mathbf{x} = \mathbf{x} + \mathbf{w}$ 
2   for each ( $i = 0, 1, \dots, R-1$ ) if ( $x_i \geq p-1$ )  $x_i = x_i - (p-1)$ 
3   load  $s(\mathbf{x})$  from the data memory
4    $\mathbf{y} = \mathbf{b} + s(\mathbf{x})$ 
5   for each ( $i = 0, 1, \dots, R-1$ ) if ( $y_i < N$ ) send  $y_i$  to the address queue
6   if ((column_counter--)  $\neq$  0) goto loop
-----

```

[0074]

TABLE 2

```

-----
0   column_counter = C-1
loop:
1    $\mathbf{x} = \mathbf{x} + \mathbf{w}$ 
2   for each ( $i = 0, 1, \dots, R-1$ ) if ( $x_i \geq 2^n$ )  $x_i = x_i - 2^n$ 
3    $\mathbf{y} = \mathbf{b} + \mathbf{x}$ 
4   for each ( $i = 0, 1, \dots, R-1$ ) if ( $y_i < N$ ) send  $y_i$  to the address queue
5   if ((column_counter--)  $\neq$  0) goto loop
-----

```

4. The attempt to incorporate subject matter (a formula for calculating  $s(x)$ ) into this application by reference to the cdma2000 standard is ineffective because incorporation of essential subject matter by reference to a non-patent publication is not appropriate.

***Claim Rejections - 35 USC § 112***

5. The following is a quotation of the first paragraph of 35 U.S.C. 112:

The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

6. Claims 3, 7, 10 and 20 are rejected under 35 U.S.C. 112, first paragraph, as based on a disclosure which is not enabling.

A formula for calculating a vector  $s(x)$  "from  $x$ ", which vector  $s(x)$  apparently corresponds to the claimed "vector that is calculated from the cumulative vector" where the "cumulative vector" is  $x$ , critical or essential to the practice of the invention but not included in the claims, is not enabled by the disclosure.

See *In re Mayhew*, 527 F.2d 1229, 188 USPQ 356 (CCPA 1976)..

7. The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

8. Claims 1-20 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.



Claims 1-4, 7-11, 15, 16 and 18-20 contain numerous errors and apparently should be amended as suggested below:

1. An interleaver for rearranging sequences of data block blocks in a data processing system, the interleaver comprising:
  - a preprocessor for preparing seed variables that ~~varies~~ vary according to the interleaving method of each of a plurality of standard standards and bit rate rates; and
  - an address generator means for generating an interleaved address on the fly using the seed variables.
2. The interleaver of claim 1, wherein the number of seed variables is less than the size of block a data block.
3. The interleaver of claim 1, wherein the seed variables include a base column vector, an increment column vector, a cumulative column vector, and a modulo base, the number of elements of all three column vectors is equal to the number of rows of the interleaver, and the elements of the column vectors are arranged by inter-row permuted permutation in advance at the preprocessing, and wherein the cumulative column vector is updated by adding the increment vector to an old cumulative vector, ~~after then~~ the interleaved addresses for one column are generated by adding the base vector and a vector that is calculated from the cumulative vector, and wherein if elements of the updated cumulative column vector are larger than the modulo base, ~~the elements of the updated cumulative column vector is subtracted~~ are reduced by the modulo base.
4. A turbo decoding system comprising:
  - a block interleaver;
  - an address queue for storing the a generated interleaved address that is equal or smaller than the size of a block of data;
  - an SISO decoder performing recursive decoding and calculating a log likelihood ratio; and
  - an LLR memory connected to the SISO decoder and storing the log likelihood ratio,wherein the block interleaver comprises a preprocessor for preparing seed variables and an address generator for generating an interleaved address on the fly using the seed variables, wherein the SISO decoder operates by accessing the data block and the log likelihood ratio in a sequential order and in an interleaved order alternately by the generated interleaved address.
7. The turbo decoding system of claim 4, wherein the seed variables include a base column vector, an increment column vector, a cumulative column vector, and a modulo base, the number of elements of all three column vectors is equal

to the number of rows of the block interleaver, and the elements of the column vectors are arranged by inter-row permuted permutation, and wherein the cumulative column vector is updated by adding the increment vector to the old cumulative vector, after then the interleaved addresses for one column are generated by adding the base vector and a vector that is calculated from the cumulative vector, and wherein if elements of the updated cumulative column vector are larger than the modulo base, ~~the elements of the updated cumulative column vector is subtracted~~ are reduced by the modulo base.

8. A turbo decoding system comprising:
  - a processor for generating interleaved addresses;
  - an address queue for storing the interleaved addresses;
  - a buffer memory block including an LLR memory for storing a log likelihood ratio and a plurality of memory blocks for storing soft inputs; and
  - an SISO decoder connected to the buffer memory block, the SISO decoder including an ACSA network for calculating the a log likelihood ratio recursively from soft inputs and the log likelihood provided by the LLR memory and a plurality of memory blocks connected to the ACSA network.
9. The turbo decoding system of claim 8, wherein the processor prepares seed variables when an interleaver structure changes due to the a change of the a coding standard or bit rate, and generates the interleaved addresses column by column using the seed variables when the interleaved addresses are required.
10. ~~The interleaving method~~ turbo decoding system of claim 9, wherein the seed variables include a base column vector, an increment column vector, a cumulative column vector, and a modulo base, the number of elements of all three column vectors is equal to the number of rows of the interleaver, and the elements of the column vectors are arranged by inter-row permuted permutation, and wherein the cumulative column vector is updated by adding the increment vector to the old cumulative vector, after then the interleaved addresses for one column are generated by adding the base vector and a vector that is calculated from the cumulative vector, and wherein if elements of the updated cumulative column vector are larger than the modulo base, ~~the elements of the updated cumulative column vector is subtracted~~ are reduced by the modulo base.
11. The turbo decoding system of claim 8, wherein the SISO decoder supports a Viterbi decoding mode, wherein in Viterbi decoding mode, the ACSA network performs Viterbi recursion, the LLR memory stores traceback information outputted by the ACSA network, the processor ~~processes~~ performs traceback from the traceback information read from the LLR memory, and one of the memory blocks of the SISO decoder stores a path metric outputted by the ACSA network.

15. The turbo decoding system of claim 13, wherein the SIMD processor includes five processing elements for parallel processing, and wherein one of five processing elements controls the other four processing elements, processes scalar operation, and fetches, decodes, and executes instructions including control and multi-cycle scalar instructions, and wherein the other four processing elements only ~~executes~~ execute SIMD instructions.

16. The turbo decoding system of claim 14, wherein the SIMD processor includes five processing elements, and wherein one of five processing elements controls the other four processing elements, processes scalar operation, and fetches, decodes, and executes instructions including control and multi-cycle scalar instructions, and wherein the other four processing elements only ~~executes~~ execute SIMD instructions.

18. An interleaving method for rearranging a data block in a data communication system, comprising:  
preparing seed variables; and  
generating interleaved addresses column by column using the seed variables.

19. The interleaving method of claim 18, wherein the seed variables are prepared when an interleaver structure changes due to ~~the~~ a change of the a coding standard or bit rate, and ~~generates~~ the interleaved addresses are generated column by column using the seed variables when the interleaved addresses are required.

20. The interleaving method of claim 19, wherein the seed variables include a base column vector, an increment column vector, a cumulative column vector, and a modulo base, the number of elements of all three column vectors is equal to the number of rows of the interleaver, and the elements of the column vectors are arranged by inter-row ~~permutated~~ permutation, and wherein the cumulative column vector is updated by adding the increment vector to the old cumulative vector, ~~after~~ then the interleaved addresses for one column are generated by adding the base vector and a vector that is calculated from the cumulative vector, and wherein if elements of the updated cumulative column vector are larger than the modulo base, ~~the elements of the updated cumulative column vector is subtracted~~ are reduced by the modulo base.

***Claim Rejections - 35 USC § 102***

9. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

10. Claims 1, 2, 18 and 19 are rejected under 35 U.S.C. 102(e) as being anticipated by U.S. Patent No. 6,721,908 to Kim *et al* (hereafter "Kim").

Regarding claims 1 and 2, Kim discloses arrangements for interleaving and deinterleaving data for a turbo code decoder operating "on the fly" in accordance with plural standards (IMT-2000, UMTS, CDMA-200) and their plural bit rates. Kim shows interleaver logic (Fig. 10) based on a "preprocessor for preparing seed variables that varies (sic – vary?) according to the interleaving method of each standard and bit rate" (initial seed #0, #1, etc.), and an "address generator means for generating an interleaved address on the fly using the seed variables" (mux 15, counter 19, etc.).

Regarding claims 18 and 19, Kim's low address portion and high address portion correspond to row addresses and column addresses, and by generating column addresses "on the fly," Kim generates addresses "column by column."

***Claim Rejections - 35 USC § 103***

11. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

12. Claims 4, 8, 9 and 11 are rejected under 35 U.S.C. 103(a) as being unpatentable over U.S. Patent No. 7,058,874 to Zhou (hereafter "Zhou").

Zhou discloses an interleaver for a "turbo decoding system." Zhou's interleaver comprises "a processor for generating interleaved addresses" (10). Zhou provides an address cache (tiwina, tiwinb) serving as "an address queue for storing the interleaved addresses." Zhou further provides "a buffer memory block including an LLR memory for storing (a) log likelihood ratio" (110).

Regarding claim 4, Zhou's address cache also serves as an "address queue for storing (a) generated interleaved address equal or smaller than the size of (a) block (of) data." Zhou's interleaver is a "block interleaver" that comprises a "preprocessor for preparing seed variables" and an "address generator for generating an interleaved address on the fly using the seed variables." Zhou's turbo code decoder operates by "accessing the data block and the log likelihood ratio in a sequential order and in an interleaved order alternately by the generated interleaved address."

As Zhou does not show the entire turbo code decoder, Zhou does not show an "SISO decoder performing recursive decoding and calculating (a) log likelihood ratio" connected to the "LLR memory" (110).

Official Notice is given that implementing a turbo code decoder with a "SISO decoder performing recursive decoding and calculating (a) log likelihood ratio" connected to an "LLR memory" was already conventional at the time the invention was made. It would have been obvious to a person having ordinary skill in the art at the time the invention was made to implement Zhou's turbo code decoder with a "SISO decoder performing recursive decoding and calculating (a) log likelihood ratio" connected to an "LLR memory." Such an implementation would have been obvious because implementing a turbo code decoder with a "SISO decoder performing recursive decoding and calculating (a) log likelihood ratio" connected to an "LLR memory" was already conventional.

Regarding claim 8, as Zhou does not show the entire turbo code decoder, Zhou does not show "a plurality of memory blocks for storing soft inputs" or "an SISO decoder connected to the buffer memory block, the SISO decoder including (an) ACSA network for calculating the log likelihood ratio recursively from soft inputs and the log likelihood provided by the LLR memory and a plurality of memory blocks connected to the ACSA network."

Official Notice is given that implementing a turbo code decoder with memory blocks for storing soft inputs, an SISO decoder connected to a buffer memory block and including an ACSA network for calculating the log likelihood ratios recursively from soft inputs and the log likelihood provided by the LLR memory, and a plurality of memory blocks connected to the ACSA network was already known at the time the invention was made. It would have been obvious to a person having ordinary skill in the art at the

Art Unit: 2133

time the invention was made to implement Zhou's turbo code decoder with the turbo encoder elements recited in claim 8. Such an implementation would have been obvious because implementing a turbo code decoder with memory blocks for storing soft inputs, an SISO decoder connected to a buffer memory block and including an ACSA network for calculating the log likelihood ratios recursively from soft inputs and the log likelihood provided by the LLR memory, and a plurality of memory blocks connected to the ACSA network was already known.

Regarding claim 11, as Zhou does not show the entire turbo code decoder, Zhou does not show a turbo code decoder that supports a Viterbi decoding mode.

Official Notice is given that the usefulness of implementing a turbo code decoder such that it has a Viterbi decoding mode was already known at the time the invention was made. It would have been obvious to a person having ordinary skill in the art at the time the invention was made to implement Zhou's turbo code decoder such that it has a Viterbi decoding mode in accordance with claim 11. Such an implementation would have been obvious because the usefulness of implementing a turbo code decoder such that it has a Viterbi decoding mode was already known.

***Allowable Subject Matter***

13. Claims 3, 5-7, 10, 12-17 and 20 would be allowable if rewritten to overcome the rejection(s) under 35 U.S.C. 112, 2nd paragraph, set forth in this Office action and to include all of the limitations of the base claim and any intervening claims.

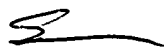
***Conclusion***

14. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

15. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Stephen M. Baker whose telephone number is (571) 272-3814. The examiner can normally be reached on Monday-Friday (11:00 AM - 7:30 PM).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Albert DeCady can be reached on (571) 272-3819. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

  
Stephen M. Baker  
Primary Examiner  
Art Unit 2133

smb